
Proceedings Paper:
Swallow, David Mark orcid.org/0000-0001-7405-2370, Petrie, Helen orcid.org/0000-0002-0100-9846 and Power, Christopher Douglas orcid.org/0000-0001-9486-8043 (2016) Understanding and supporting web developers: : design and evaluation of a web accessibility information resource (WebAIR). In: Universal Design 2016:Learning from the past, designing for the future (Proceedings of the 3rd International Conference on Universal Design, UD2016). IOS Press , Amsterdam , pp. 482-491.

Understanding and Supporting Web Developers: Design and Evaluation of a Web Accessibility Information Resource (WebAIR)

David SWALLOW[1], Helen PETRIE, and Christopher POWER
Human Computer Interaction Research Group, Department of Computer Science, University of York, UK

Abstract. This paper describes the design and evaluation of a Web Accessibility Information Resource (WebAIR) for supporting web developers to create and evaluate accessible websites. WebAIR was designed with web developers in mind, recognising their current working practices and acknowledging their existing understanding of web accessibility. We conducted an evaluation with 32 professional web developers in which they used either WebAIR or an existing accessibility information resource, the Web Content Accessibility Guidelines, to identify accessibility problems. The findings indicate that several design decisions made in relation to the language, organisation, and volume of WebAIR were effective in supporting web developers to undertake web accessibility evaluations.

Keywords. web accessibility, evaluation, web developers, web accessibility guidelines, web accessibility information resources

1. Introduction

Web developers have an obligation to develop websites that are accessible and usable by the broadest range of users, including people with disabilities. Over the last 20 years, there have been various initiatives to support, encourage and compel web developers to fulfil this obligation. These initiatives include projects, working groups, and task forces, such as the World Wide Web Consortium's (W3C) Web Accessibility Initiative (WAI)[2], the WAI's Education & Outreach Working Group (EOWG)[3], and the (now defunct) Web Standards Project's (WaSP) Accessibility Task Force (ATF)[4]. Other initiatives include accessibility legislation, such as the UK's Equality Act [1] and the US's Americans with Disabilities Act (ADA) [2]. These efforts have resulted in a well-established body of accessibility information, often presented in the form of a set of guidelines or recommendations, such as the US government's Section 508 standards [3] and the WAI's Web Content Accessibility Guidelines (WCAG) [4].

[1] Corresponding Author, Human Computer Interaction Research Group, Department of Computer Science, University of York, York YO10 5GH, UK; E-mail: david.swallow@york.ac.uk
[2] https://www.w3.org/WAI/
[3] https://www.w3.org/WAI/EO/
[4] http://www.webstandards.org/action/atf/

Despite these well-intentioned initiatives and the wide availability of accessibility information, web developers still struggle to create accessible websites. Evidence from a substantial number of web accessibility evaluation studies over the last 20 years indicates that the accessibility of websites has barely improved during this period and, according to certain studies (e.g. [5][6]), has worsened. We can partly attribute this disappointing trend to external factors that are out of web developers' control, such as client and organisational attitudes to web accessibility as well as difficulties in enforcing accessibility legislation [7]. Much of web developers' struggle to create accessible websites, however, can be attributed to difficulties in understanding and interpreting the accessibility information provided by tools, guidelines and resources.

Several studies, and much anecdotal evidence, suggest that web developers find accessibility information confusing. A study of accessibility evaluation tools found they offered inadequate support to web developers in checking the accessibility of their web resources. Further, the tools did little to enhance web developers' understanding of accessibility issues [8]. A study of web developer attitudes to web accessibility also highlighted difficulties in interpreting the output of accessibility evaluation tools [9]. A study of WCAG with 35 student web developers found that, for many of the guidelines, web developers were unable to come to an 80% level of agreement about whether a problem was present in a webpage [10]. A study of 17 students taking part in a web accessibility course concluded that WCAG is "far from testable for beginners" (p.8). The authors attributed this to: difficulties in comprehending the language used in the guidelines; a lack of knowledge that is required to correctly evaluate the guidelines; and a reluctance to spend a lot of effort evaluating the guidelines [11]. These studies highlight the significant problems that web developers encounter in understanding and interpreting the accessibility information provided by tools, guidelines and resources.

Efforts to deliver accessibility information to web developers in a more meaningful way have largely focused on managing and administering existing sets of guidelines. For example, the MAGENTA tool [12] assists web developers in defining, handling and checking multiple sets of accessibility guidelines. It performs semi-automatic accessibility evaluations and provides web developers with advice on how to address accessibility issues. The Accessibility Evaluation Assistant (AEA) [13] supports novice accessibility evaluators in conducting accessibility evaluations. It presents a series of tailored checkpoints by filtering web accessibility guidelines according to particular user groups or types of web content. The WAI also provide a range of support materials[5] aimed at making WCAG more digestible and comprehensible to web developers. None of these commendable efforts, however, appear to have considered web developers' working practices or explored their existing knowledge and understanding of web accessibility.

Building on these and our own studies, we developed and evaluated the Web Accessibility Information Resource (WebAIR) to support web developers in creating and evaluating accessible websites. It presents web developers with a digestible amount of accessibility information that is written in web development-oriented language and organised around their existing workflows. This paper summarises our investigations into the working practices of web developers, identifies the problems that web developers encounter in attempting to create and evaluate accessible websites, describes the design and implementation of WebAIR, and presents the results of a series of evaluations with professional web developers.

[5] https://www.w3.org/WAI/eval/Overview.html

2. Web Development In Context

To understand why web developers struggle to make websites accessible and how best to support them, we conducted an initial investigation into the working practices of 13 professional web developers from the UK, Ireland and Italy. This research was undertaken as part of the i2Web project[6], which aimed to create tools for developing and evaluating accessible web applications.

The web developers in this investigation had between 1 and 15 years' experience of web development, with an average of 9 years. Six participants worked for large enterprises (250+ employees), six worked for SMEs (< 250 employees) and one participant was a self-employed freelancer. The investigation drew upon the rich ethnographic methodology of contextual inquiry [14] to interview and observe the web developers carrying out their own work in their own work environment.

The findings of this investigation indicated that despite being genuinely interested in web accessibility, web developers still struggle to develop accessible websites [15]. Web developers are hindered, not by limited awareness or concern, but by a lack of knowledge and practical guidance on how to make websites accessible. Existing tools, guidelines and resources are letting web developers down by not providing them with the support and information they need. Three key themes emerged as to why developers struggle with accessible web development.

The first theme – *Language* – represents how existing tools, guidelines and resources do not speak the language of web developers. Instead, they tend to rely upon vague statements that assume web developers are familiar with domain-specific concepts of web accessibility. These statements include undefined directives, such as "provide users enough time to read and use content" as well as optional warnings, such as "you may need to check the alt description of this image", both of which web developers find unhelpful and off-putting.

The second theme – *Organisation* – represents how existing tools, guidelines and resources are often organised in different ways to how web developers approach the creation of websites. While web developers' practices are typically related to the code they are working on, existing tools, guidelines and resources tend to use domain-specific groupings (such as "Perceivable" or "Operable") that web developers find unfamiliar. Consequently, the information appears abstract, arbitrary and unrelated either to the web developers' work or to the people it is intended to benefit.

The third theme – *Volume* – represents how existing tools, guidelines and resources tend to present web developers with too much information and too many items to test at once, resulting in information and procedural overload. Though web developers acknowledge the substantial amount of work involved in making websites accessible, they find the amount of information presented by existing tools, guidelines and resources overwhelming and the number of items to test unrealistic.

This contextual inquiry investigation determined that without access to a appropriate amount of clear, concise and precise accessibility information that they can easily interrogate, understand, and apply to their work, web developers struggle to determine the accessibility of their websites. This manifests in low confidence and self-efficacy with regards to web accessibility; an over-reliance on automated accessibility evaluation tools and dedicated accessibility experts; and, ultimately, a failure to integrate accessible web development practices into their existing workflows.

[6] http://i2web.eu/

3. Design of WebAIR

To address the problems that web developers encounter in making websites accessible, we developed the Web Accessibility Information Resource (WebAIR). WebAIR is intended to help web developers learn about web accessibility and support them in creating and evaluating accessible websites. It presents a digestible amount of accessibility information that is written in web development-oriented language and organised around the existing workflows and working practices of web developers.

We made several key design decisions in the development of WebAIR. Each design decision addresses one or more of the themes that emerged from the previous investigation concerning why web developers struggle with accessible web development.

3.1. Language

One of the themes that emerged from the previous investigation was that existing tools, guidelines and resources do not speak the language of web developers.

We addressed this theme in WebAIR by phrasing accessibility issues as concrete, objective questions for web developers to use in checking their web content. Each question asks web developers whether they have completed a particular task. For example, keyboard accessibility of websites is determined by asking web developers: "Can you successfully access all links using the keyboard?" Similarly, to avoid ambiguities around form submission, WebAIR asks web developers: "Do you provide feedback when a form has been submitted successfully?" We constructed the questions to avoid the domain-specific language of web accessibility. Instead, they incorporate web development terms or refer to specific user actions in the interface.

To ensure that WebAIR encapsulates coverage of existing accessibility guidelines, we undertook a mapping from the WCAG 2.0 [4] Success Criteria and Techniques onto each question. Each of the 205 questions in WebAIR maps to one or more WCAG 2.0 Success Criteria and Techniques.

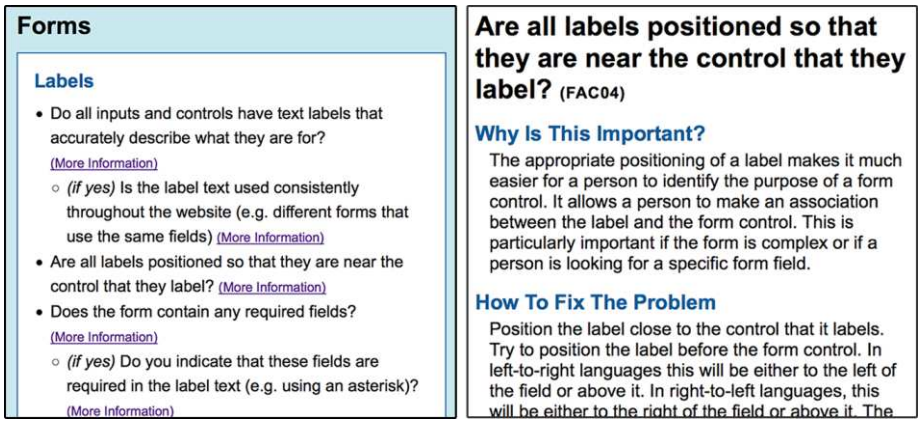

Figure 1. Example screenshots of WebAIR, showing a selection of questions in the 'Forms' category (left) and a 'More Information' page relating to a question on form control labelling (right).

3.2. Organisation

Another theme that emerged from the previous investigation was that existing accessibility information resources are often organised in different ways to how web developers approach the creation of websites.

We addressed this theme in WebAIR by orienting the resource towards web developers' workflows, specifically their tendency to structure their work according to the types of web content on which they are working, either during web development or evaluation. We organised the previously described questions in WebAIR according to ten different types of web content: forms, links, tables, images, text, multimedia, interactive content, time-limited content, within page content, and between page content. Each content type category contains several sub-categories to further classify the questions. For example, the category relating to forms contains the sub-categories: labelling; grouping; navigating form fields; completing forms; and errors. By organising the content in this way, we aimed to provide web developers with a concrete categorisation of information that they can more easily apply to their work.

3.3. Volume

Another theme that emerged from the previous investigation was that existing tools, guidelines and resources tend to present web developers with too much information and too many items to test at once, resulting in information and procedural overload.

We addressed this theme in WebAIR by creating a 'More Information' page for each question. These provide a brief (typically one paragraph) rationale as well as practical instructions on how to answer the question and address the accessibility issue. We also included external links to further reading and information, where available.

The intention of this design decision was to provide web developers with just-in-time training in web accessibility concepts, allowing them to gradually learn about the domain of web accessibility while they undertake testing. Further, by offering the rationale behind each accessibility recommendation, we aimed to tackle an important issue that emerged in the contextual inquiry investigation: that web developers sometimes feel they are blithely following seemingly arbitrary guidelines.

In addition to presenting a reduced amount of accessibility information, each 'More Information' page includes only one example solution drawn from the numerous techniques provided in WCAG 2.0. While not as comprehensive in the variety of ways in which web accessibility may be achieved, we intended this reduction to address the issue of information and procedural overload problems that web developers currently encounter in attempting to apply web accessibility guidance to their web content.

The accessibility information that WebAIR contains may be delivered as a standalone multi-page website (similar to WCAG 2.0) or it may be embedded into web development and evaluation tools (as in the i2Web project). For the purpose of evaluating WebAIR, we created a standalone multi-page website[7], comprising a main index page listing all of the questions categorised by content type. Each question linked to a 'More Information' page containing the accessibility information. To control for the appearance of WebAIR, we present it using a similar colour scheme and style to the W3C standard template (see Figure 1 for example screenshots of WebAIR).

[7] WebAIR is available at: http://www.cs.york.ac.uk/hci/webair/

4. Initial Evaluation and Revision of WebAIR

To determine whether the design decisions embodied in WebAIR improve access to accessibility information for web developers, we undertook a within-participants evaluation with 26 web developers. We gave participants the opportunity to use both WebAIR and WCAG 2.0 to undertake accessibility testing.

The results of this evaluation (described in more detail in [16]) indicated that the design decisions relating to the language, organisation and volume of WebAIR were largely successful in improving access to web accessibility information. Though the majority of participants found WebAIR easier to use and understand than WCAG 2.0, some considered the language in WebAIR too basic for web developers. Others felt that some of the WebAIR content categories were unusually worded and difficult to apply to their work. Also, despite WebAIR being much smaller than WCAG 2.0, some participants considered it still too large.

In response to the outcome of this initial evaluation, we substantially revised WebAIR, re-writing much of the content to make it more relevant to web developers, renaming some of the content categories, and merging or removing some of the questions to further reduce the amount of information and number of items to test. The revised version of WebAIR contains 154 questions (a 25% reduction from the previous version) organised according to eight different types of web content: forms, links, tables, images, text, audio & video, time limits, and navigation.

5. Further Evaluation of WebAIR with Professional Web Developers

To determine the effectiveness of the revised version of WebAIR in supporting website accessibility evaluation, we undertook a between-participants evaluation with 32 web developers from Australia, the Netherlands, and the UK. Participants used either WebAIR or WCAG 2.0 to identify accessibility problems in the homepage of a custom-built website. In addition to measuring participants' performance in this task, we determined the effectiveness of the two resources through a combination of rating items and self-reported usage data elicited using a think aloud protocol.

5.1. Participants

The 32 web developers had between 1 and 20 years' experience of web development, with an average of 10 years. Eight participants worked for large enterprises, eighteen worked for SMEs and six participants were self-employed freelancers.

Prior to the evaluation, participants reported being fairly familiar with web accessibility (mean rating: 3.9 out of 5; SD: 0.71) and moderately familiar with WCAG (mean: 3.1 out of 5; SD: 1.13) (both scales: 1 = Not at all familiar, 5 = Very familiar).

5.2. Materials

Participants undertook the evaluations using WebAIR and the WCAG 2.0 guidelines.

For WebAIR, participants used the revised version of the resource described in the previous section, comprising 154 questions organised according to 8 web content types.

For WCAG 2.0, participants used the official documentation provided by the WAI [4]. This multi-page website provides information on the 12 Guidelines and 61 Success Criteria (SC) of WCAG 2.0. Two pages accompany each SC: one on 'How to Meet' that specific SC, serving as a quick reference for developers, and one on 'Understanding' that SC, providing a more detailed explanation.

Participants used these resources to identify and resolve accessibility problems in the homepage of "Eat My Goal!" [8], a custom-built football news website, which we purposely designed to incorporate a range of accessibility problems.

5.3. Procedure

To afford greater flexibility in recruiting participants and allow participants to undertake the evaluations in a more realistic environment, we conducted half of the evaluations in person, at a location convenient to each participant (e.g. their workplace, home or a café) and half remotely, using video conferencing software.

Following a brief introduction to their allocated accessibility information resource (WebAIR or WCAG 2.0), we gave participants 35 minutes to identify accessibility problems in the evaluation website. We explained to participants that that it was up to them how they approached the task and how they made use of the resource. We instructed them, however, that they should refrain from using automated accessibility evaluation tools. Irrespective of how participants approached the task, we asked them to describe what they were doing and thinking using a concurrent think aloud protocol.

Immediately after the task, we asked participants to complete a short questionnaire comprising 9 five-point Likert items: Usefulness (1 = very low, 5 = very high), Ease of Use (1 = very difficult, 5 = very easy), Navigability (1 = very low, 5 = very high), Understandability (1 = very low, 5 = very high), Completeness (1 = very low, 5 = very high), Amount of Information Provided (1 = far too little, 5 = far too much), Number of Items to Test (1 = far too few, 5 = far too many), Organisation (1 = very unclear, 5 = very clear), and Likelihood of Using the web accessibility information resource (1 = very unlikely, 5 = very likely). Evaluation sessions lasted approximately 45 minutes.

5.4. Results

5.4.1. Website Accessibility Evaluation

The evaluation webpage presented 45 distinct accessibility problems for participants to identify. Participants identified between 4 and 30 accessibility problems in each 35-minute evaluation session (overall mean: 15.5 problems; SD: 6.02). This amount represents just over a third (34%) of the potential accessibility problems in the webpage. WebAIR users identified between 8 and 30 accessibility problems in each evaluation session (mean: 17.8 problems; SD = 6.19). WCAG 2.0 users identified between 4 and 22 accessibility problems in each evaluation session (mean: 13.2 problems; SD = 5.01).

We conducted an independent-samples t-test to compare the number of problems identified by WebAIR and WCAG 2.0 users. This test showed a significant effect of the type of resource, $t(30) = 2.32$, $p < .05$; $d = 0.82$. This result indicates that participants identified a significantly greater number of problems using WebAIR to evaluate the accessibility of the webpage than using WCAG 2.0.

[8] https://www.cs.york.ac.uk/hci/webair_studies/eatmygoal/

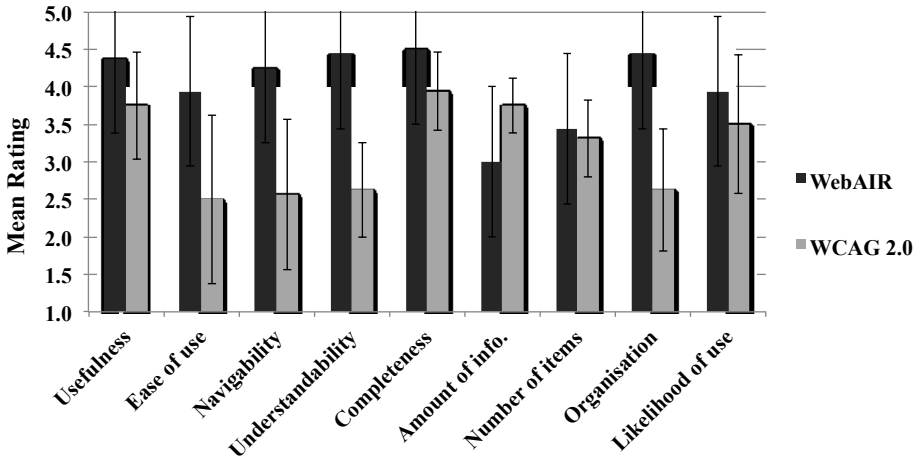

Figure 2. Mean ratings of WebAIR and WCAG 2.0 on a range of attributes

5.4.2. Accessibility Information Resource Ratings

Figure 2 shows the mean ratings for WebAIR and WCAG 2.0 on the Likert rating items. A one-way MANOVA was used to investigate the effect of the type of resource on the Likert ratings. This revealed a significant multivariate effect of type of resource ($F = 8.11$, $df = 9,22$, $p < .005$). Overall, the web developers rated the WebAIR resource significantly highly than the WCAG 2.0 resource (mean for WebAIR: 4.03, SD: 0.73; mean for WCAG 2.0: 3.17, SD: 0.95).

The univariate tests showed significant effects of type of resource on all of the Likert ratings except completeness and the number of items to test. WebAIR was rated significantly higher than WCAG 2.0 on ease of use ($F = 14.20$, $p < .05$, all tests $df = 1,30$), navigability ($F = 27.41$, $p < .05$), understandability ($F = 40.05$, $p < .05$), completeness ($F = 5.08$, $p < .05$), and organisation ($F = 26.56$, $p < .05$). WCAG 2.0 was rated significantly higher than WebAIR on amount of information ($F = 10.39$, $p < .05$). Though WebAIR was rated slightly higher than WCAG 2.0 on usefulness, the number of items to test, and the likelihood of using, there was no significant difference between the ratings for these items.

5.4.3. Observations of Accessibility Information Resource Usage

Based on the participants' self-reported usage data and our own observations, we determined how they approached the website accessibility evaluation task and whether the two resources were effective in supporting their behaviour.

Using either WebAIR or WCAG 2.0, we observed a noticeable difference in how participants attempted the task. Some participants began from the evaluation website, inspecting both the content and the underlying code for potential problems, before consulting the resource. Those who did this tended to use the resource as a reference tool for looking up and learning about specific accessibility issues. Others started from the resource, working through the accessibility information it contains and applying it to the evaluation website. Those who did this tended to treat the resource as a prompt or checklist, against which they could evaluate the website. The effectiveness of the

two resources in supporting each approach, however, appeared to be mediated by the participant's familiarity with web accessibility.

Participants who were less familiar with web accessibility and who approached the task by first inspecting the website, struggled to use WCAG 2.0 for this purpose. They found its language unclear, its organisation confusing, and its volume of information overwhelming, making the resource difficult to interrogate and its guidance difficult to apply. Conversely, WebAIR users praised the familiarity of its language and the clarity of its organisation, which they felt make it an effective reference tool.

Participants who approached the task by first inspecting the website but who were more familiar with web accessibility tended only to consult the resources to look up unfamiliar accessibility issues, or to bolster existing knowledge. Participants using WCAG 2.0 in this regard found it comprehensive and informative, whereas WebAIR users felt it does not provide sufficient detail about some accessibility issues.

Participants who were less familiar with web accessibility and who approached the task by first consulting the resource, struggled to use WCAG 2.0 for this purpose. They felt the organisation of the resource provides no clear order to follow and, because specific authoring practices are buried deep in its organisation, it offers no clear calls to action. Conversely, WebAIR users felt that its content-oriented organisation provides a clear order to follow and its question-based phrasing offered clear calls to action.

Finally, participants who approached the task by first consulting the resource but who were more familiar with web accessibility tended to use the resources more as springboards to their own knowledge. Participants using either resource in this regard rarely looked beyond than the top-level Success Criteria or questions. However, due to participants either misinterpreting issues (e.g. one participant incorrectly assumed that the WCAG 2.0 Guideline 'Readable' referred to the presentation and not the comprehensibility of text content) or applying their own incorrect or out-dated knowledge, this approach was not always successful. Such misinterpretation was more noticeable in WebAIR users, who were perhaps lulled into a false sense of familiarity by the web development-oriented language and terminology.

6. Discussion and Conclusions

WebAIR was intended to help web developers learn about web accessibility and support them in creating and evaluating accessible websites. It was designed (and re-designed) with web developers in mind, recognising their current workflows and working practices and acknowledging their existing understanding of web accessibility.

Both the quantitative and qualitative findings of this investigation indicate that the design decisions made in relation to the language, organisation and volume of WebAIR were effective in supporting web developers to undertake web accessibility evaluations. Participants using WebAIR to evaluate the accessibility of a website identified a significantly greater number of accessibility problems than those using WCAG 2.0. They also rated WebAIR significantly more highly than WCAG 2.0 across a number of attributes, including ease of use, navigability, understandability, clarity of organisation, and – surprisingly, given the comprehensiveness of WCAG 2.0 - completeness.

Observations of participants using the two resources indicate that WebAIR is particularly effective in supporting web developers who are less familiar with web accessibility. WebAIR's web development-oriented language and organisation appears to aid comprehension and navigation in web accessibility newcomers. Its reduced

volume, however, may be insufficient for web developers who are more familiar with web accessibility and who want to learn about the subject in more depth.

Our evaluation is not without its limitations. The amount of time we gave participants to complete the task (35 minutes) was not sufficient to undertake a full accessibility evaluation – a process which, in reality, may take hours or even days. Further, in asking participants to evaluate a website that was not their own, we may have added a further degree of artificiality to the task. Nevertheless, our evaluation demonstrates the benefits of designing accessibility information resources with web developers in mind.

WebAIR is an effective accessibility information resource that provides a much-needed complement to existing tools, guidelines and resources. It delivers accessibility information to web developers in ways that play to their strengths and respect their existing workflows and working practices. Our future work will focus not only on evaluating WebAIR under more realistic conditions, but also on determining its effectiveness in supporting both the evaluation and creation of accessible websites.

References

[1] Equality Act 2010. London: HMSO.
[2] Americans with Disabilities Act (US Public Law 101--336).
[3] Section 508 of the Rehabilitation Act (29 U.S.C. 794d).
[4] Caldwell, B., Cooper, M., Reid, L.G., & Vanderheiden, G. (2008). Web Content Accessibility Guidelines (WCAG) 2.0. Retrieved June 21, 2016 from http://www.w3.org/TR/WCAG20/
[5] Hackett, S., Parmanto, B., & Zeng, X. (2005). A retrospective look at website accessibility over time. Behaviour and Information Technology, 24(6), pp.407–17.
[6] Loiacono, E.T., Romano Jr., N.C., & McCoy, S. (2009). The state of corporate website accessibility. Communications of the ACM, 52(9), pp. 128–32.
[7] Lazar, J., Dudley-Sponaugle, A. & Greenidge, K. (2004). Web accessibility: a study of webmaster perceptions. Computers in Human Behaviour, 20(2), 269 - 288.
[8] Petrie, H., King, N., Velasco, C., Gappa, H., & Nordbrock, G. (2007). The usability of accessibility evaluation tools. In Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction: Applications and Services, Beijing, China, July 22-27.
[9] Trewin, S., Cragun, B., Swart, C., Brezin, J & Richards, J. 2010. Accessibility challenges and tool features: an IBM Web developer perspective. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10). ACM, New York, NY, USA.
[10] Brajnik, G. 2009. Validity and reliability of web accessibility guidelines. In Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility (Assets '09). ACM, New York, NY, USA, 131-138.
[11] Alonso, F., Fuertes, J.L., Gonzalez, L.A. & Martinez, L. 2010. On the testability of WCAG 2.0 for beginners. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10). ACM, New York, NY, USA.
[12] Leporini, B., Paternò, F., & Scorcia, A. (2006). Flexible tool support for accessibility evaluation. Interacting with Computers, 18(5), 869-890.
[13] Bailey, C., & Pearson, E. (2010). An educational tool to support the accessibility evaluation process. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10). ACM, New York, NY, USA.
[14] Beyer, H. & Holtzblatt, K. (1997). Contextual Design: Defining Customer-Centered Systems. San Francisco: Morgan Kaufmann Publishers Inc.
[15] Petrie, H., Power, C., & Swallow, D. (2011). i2Web Deliverable 3.2: Requirements for web developers and web commissioners in ubiquitous Web 2.0 design and development. Available at: http://i2web.eu/downloads/201201_I2Web_D32.pdf
[16] Swallow, D., Power, C., Petrie, H., Bramwell-Dicks, A., Buykx, L., Velasco, C.A., Parr, A. & O Connor, J. (2014). Speaking the Language of Web Developers: Evaluation of a Web Accessibility Information Resource (WebAIR). In International Conference on Computers for Handicapped Persons (pp. 348-355). Berlin: Springer International Publishing.